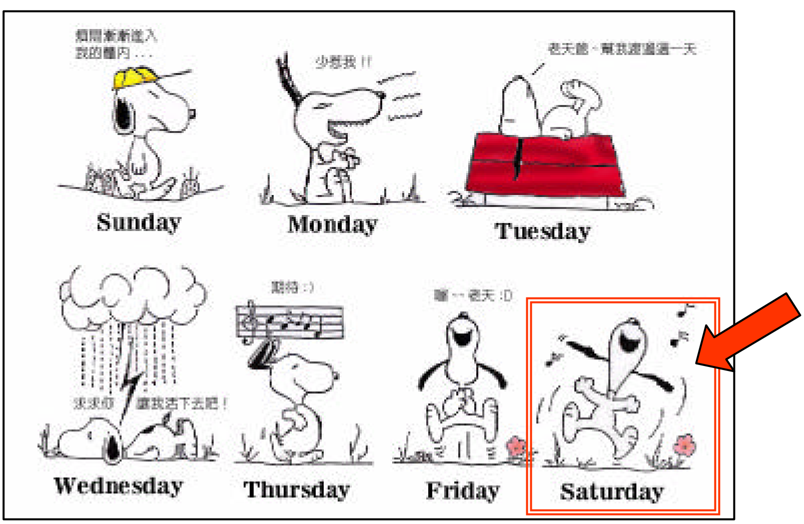
泰勒定理
在科學計算上的應用

王　偉　仲
國立高雄大學應用數學系

國際數學奧林匹亞競賽第四次培訓營，高雄大學，2004年6月19日

Isn’t it Saturday?



But, I guess you want to be the…

So, let's go!

Outline

- Brief introduction to scientific computation
- Taylor's theorem
  - The theorem
  - Finite difference
  - Initial value problem
  - Boundary value problem
  - Newton's method for root finding

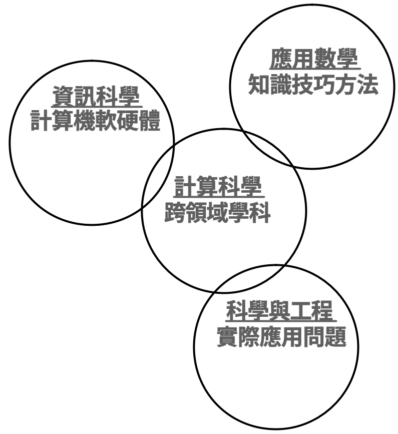
What is Scientific Computing?

- Scientific methods:
  - Theory
  - Experiment
  - Computation
- Give me a number
- How good is the number?

What is Scientific Computing? (cont.)

- **計算科學與**
  - **各科學與工程領域（實際應用問題）**
  - **應用數學（知識、技巧與方法）**
  - **以及資訊科學（計算機軟硬體）**

  彼此之間也都有相當重疊交互應用



What is Scientific Computing? (cont.)

- Being able to automate calculations with a computer
- Design and analyze algorithms for solving mathematical problems that arise in computational science and engineering
- Numerical analysis, Computational methods

What Is Scientific Computing? (cont.)

- Deals with quantities that are continuous rather than discrete
- Concerned with *approximation* and their effects
- Approximations are used not just by choice: they are inevitable in most problems

Numerical and Symbolic Calculation

Numerical	Symbolic
Involves numbers directly	Numbers are represented in an equation
$3.4^2-0.79=10.77$	$(x^2-1)/(x+1)=x-1$
3.14159265358979…	π
0.250+0.333=0.583	1/4 +1/3 = 7/12 (=0.5833…)

Numerical and Symbolic (cont.)

Numerical	Symbolic
Programming language (Fortran, C, C++, Java, Matlab)	Computer programs (Maple, Mathematica, Matlab)
Quicker / Approximation	Slower / Exact
Main focus in this course	Highlight the fundamental behavior
<u>Numbers are represented as binary (0 or 1)</u>	

Numerical Methods, Algorithms, and Computer Codes

- The goal is to find a solution of a real world problem <u>numerically</u>.
- <u>Numerical methods</u> is a mathematical description of the introduction of a idea.
- <u>Algorithm</u> is a precise sequence of actions taken to obtain a desired result.

Numerical Methods, Algorithms, and Computer Codes (cont.)

- Computer code is a translation of an algorithm into a sequence of statements in a particular computer language.

An Example

- What is the surface area of the earth?

A Solution

- Earth is modeled as sphere, an idealization of its true shape
- Mathematics gives formula A=4^R^2
- Value of radius is based on empirical measurements and previous computation

A Solution (cont.)

- Value for ^ requires truncating an infinite process
- Input data and results for arithmetic operations are rounded in computer
- Answer the question “What is the error of the computed solution?”

General Strategy

- Replace difficult problem by easier one that has same or closely related solutions.

- Complicated -> Simple
- Nonlinear -> Linear
- Infinite -> Finite
- Differential -> Algebraic

An Example of “Simplified”

KISS: Keep It Simple, Students.

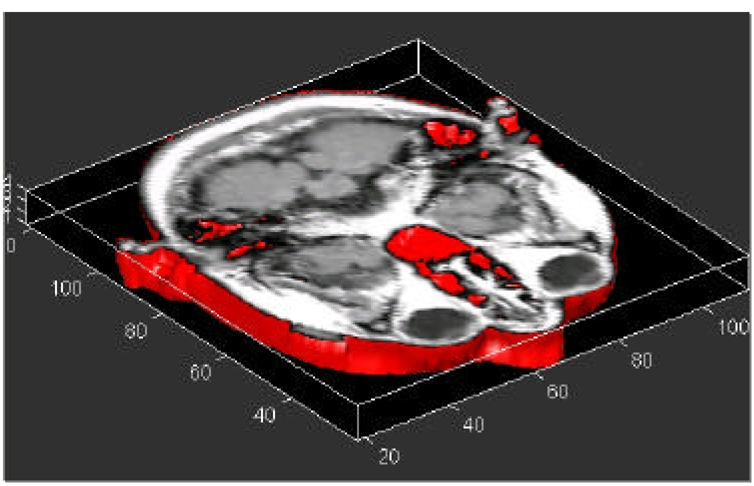
Ten Outstanding Engineering Achievements (1964-1989)

- National Academy of Engineering, 1989
  1. Microprocessor
  2. Moon landing
  3. Application satellites
  4. Computer-aided design and manufacturing
  5. Jumbo jet
  6. Advanced composite materials
  7. Computerized axial; tomography
  8. Genetic engineering
  9. Lasers
  10. Optical fibers

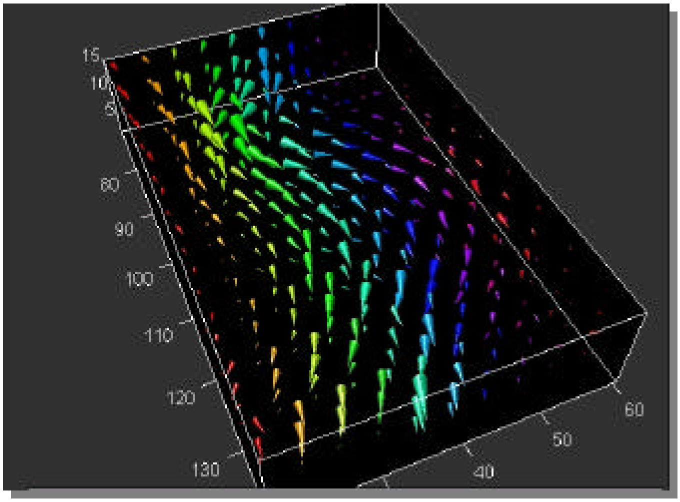
Grand Challenges

1. Prediction of weather, climate, and global change
2. Computerized speech understanding
3. Human genome project
4. Improvements in vehicle performance

核磁共振影像

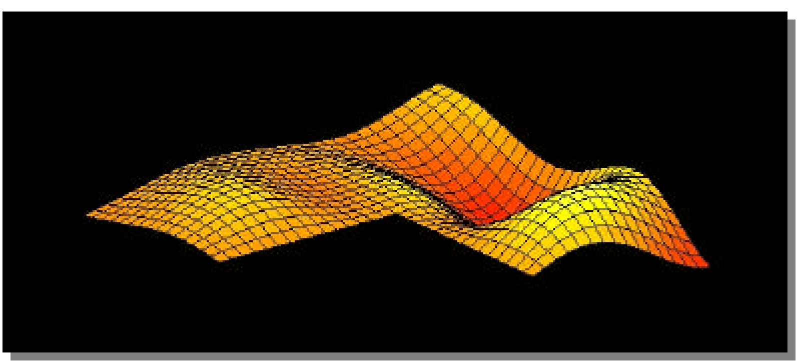


風流



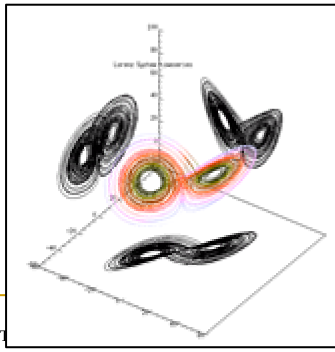
震動中的波

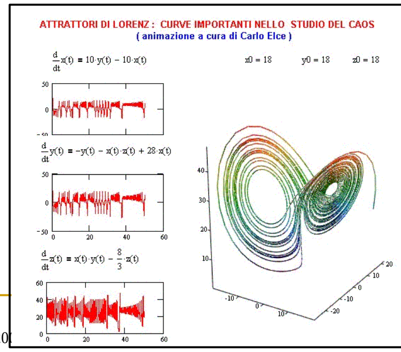
- In Matlab: demo > Graphics > Vibrating logo



Butterfly Effect

- Ed Lorenz’s paper (1963) reads “*One meteorologist remarked that if the theory were correct, one flap of a seagull's wings would be enough to alter the course of the weather forever.* “
- The talk given at American Association for the Advancement of Science in Washington, D.C. (1972) “*Predictability: Does the Flap of a Butterfly Wings in Brazil set off a Tornado in Texas?* “





Lorenz's Equations

$$\frac{dx}{dt} = \sigma(y-x)$$

$$\frac{dy}{dt} = rx-y-xz$$

$$\frac{dz}{dt} = xy-bz$$

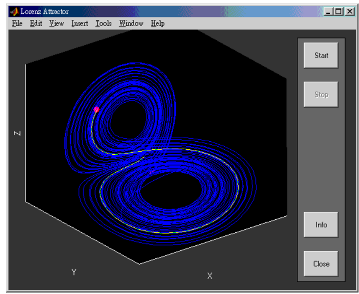
- First order coupled equation
- Commonly used constants
  - a = 10, r = 28, b= 8 / 3,
  - a = 28, r = 46.92, b = 4
- Initial of x, y, z

混沌

- In Matlab: demo > Graphics > Lorenz attractor animation

•混沌是非週期性之物理系統的演化模式。系統的演化對初始條件非常敏感。

•實例：<u>勞倫茲</u>吸引子 (Lorenz Attractor)



生物現象

- **人眼可輕易偵測移動中的物體，但難以分辨同一物體的靜止狀態**
- **對動物來說，這是重要的生存本能 (掠奪、捕食、逃避)**
- **範例: 隱藏的形狀**
  In Matlab: demo > More Examples > Hiden objects in motion

Taylor's Theorem

Brook Taylor (1685-1731)

Taylor was an ingenious and productive British mathematician.

He attended St. John's College in Cambridge, and shortly after his graduation was elected a Fellow of the Royal Society.

§1.1 Basic tools of calculus 1.1 – 11

Thm 1.1 (Taylor's Theorem with Remainder)

– It represents general functions as polynomials with a known, specified, boundable error.

– Polynomials: easy to take derivatives, integrals, ...

Let $f(x) \in C^{n+1}$ on $[a,b]$. Then

$$f(x) = P_n(x) + R_n(x)$$

$$= \sum_{k=0}^{n} \frac{(x-x_0)^k}{k!} f^{(k)}(x_0) + \frac{(x-x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\xi_x), \quad \xi_x \text{ between } x, x_0$$

DIY: For $n=3$

$\Rightarrow f(x) = ?$

12

DIY: Expand $f(x+h)$ in a Taylor series, about the point $x_0 = x$.

(p. 9) $f(x+h) = ?$

Expands at x_0.

Examples:

(p.3) (1) $e^x =$ 3

(2) $\sin x =$

(3) $\cos x =$

Question: How to evaluate the value of e^x for $x \in [-1, 1]$ with error $\leq 10^{-6}$? (Hint: p.3, p.4)

(Suppose we can do only $+, -, *, /$.)

Question: What can you say about Fig 1.1 on p.5?

Question: How can you improve the accuracy of $p_2(x) \approx e^x$ at $x = 1$?

Matlab plot Fig 1.1 on p.5.

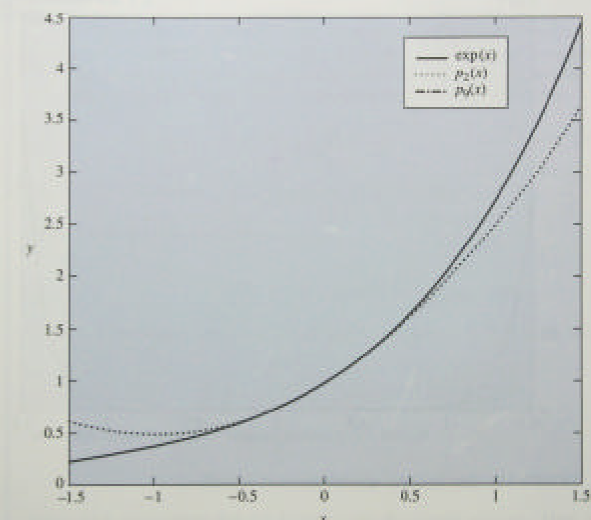


FIGURE 1.1 Taylor approximation: e^x, $p_9(x) \approx e^x$, and $p_2(x) \approx e^x$. Note that e^x and $p_9(x)$ are indistinguishable on this plot.

$$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots + \frac{1}{n!}x^n + \frac{1}{(n+1)!}x^{n+1}e^{\xi_x}$$
$$= \sum_{k=0}^{n} \frac{1}{k!}x^k + R_n(x),$$
$$\sin x = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 + \cdots + \frac{(-1)^n}{(2n+1)!}x^{2n+1} + \frac{(-1)^{n+1}}{(2n+3)!}x^{2n+3}\cos\xi_x$$
$$= \sum_{k=0}^{n} \frac{(-1)^k}{(2k+1)!}x^{2k+1} + R_n(x),$$
$$\cos x = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 + \cdots + \frac{(-1)^n}{(2n)!}x^{2n} + \frac{(-1)^{n+1}}{(2n+2)!}x^{2n+2}\cos\xi_x$$
$$= \sum_{k=0}^{n} \frac{(-1)^k}{(2k)!}x^{2k} + R_n(x).$$

Suppose we want this approximation to be accurate to within 10^{-6} in absolute error, i.e., we want

$$|e^x - p_n(x)| \le 10^{-6}$$

for *all* x in the interval $[-1,1]$. Note that if we can make $|R_n(x)| \le 10^{-6}$ for all $x \in [-1,1]$, then we will have

$$|e^x - p_n(x)| = |R_n(x)| \le 10^{-6}$$

so that the error in the approximation will be less than 10^{-6}. The best way to proceed is to create a *simple* upper bound for $|R_n(x)|$, and then use that to determine the number of terms necessary to make this upper bound less than 10^{-6}.

Thus we proceed as follows:

$$|R_n(x)| = \frac{|x^{n+1}e^{\xi_x}|}{(n+1)!} = \frac{|x|^{n+1}e^{\xi_x}}{(n+1)!}, \quad \text{because } e^z > 0 \text{ for all } z$$
$$\le \frac{e^{\xi_x}}{(n+1)!}, \quad \text{because } |x| \le 1 \text{ for all } x \in [-1,1]$$
$$\le \frac{e}{(n+1)!}, \quad \text{because } e^{\xi_x} \le e \text{ for all } x \in [-1,1].$$

Thus, if we find n such that

$$\frac{1}{(n+1)!}e \le 10^{-6}$$

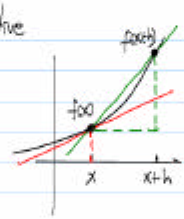
then we will have

$$|e^x - p_n(x)| = |R_n(x)| \le \frac{1}{(n+1)!}e \le 10^{-6}$$

and we will *know* that the error is less than th...

Finite difference to the derivative

§ 22 Difference approximations to the derivative



Goal: to evaluate $f'(x)$

Calculus: $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad \text{for small } h$$

$f'(x)$: 切線斜率 ; $\frac{f(x+h) - f(x)}{h}$: 割線斜率

Taylor: $f(x+h) = f(x) + h f'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(\xi_x)$

$$\Rightarrow f'(x) = \frac{f(x+h) - f(x)}{h} + \frac{1}{2} h f''(\xi_x)$$

$$= \frac{f(x+h) - f(x)}{h} + O(h)$$

- One-sided difference 

- First order finite difference approximation

12

$$f(x+h) = f(x) + h f'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(\xi_1)$$

$$-)\quad f(x-h) = f(x) - h f'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(\xi_2)$$

$$f(x+h) - f(x-h) = 2h f'(x) + \frac{h^3}{6}\left(f'''(\xi_1) - f'''(\xi_2)\right)$$

$$\Rightarrow f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \left(-\frac{1}{6}\right) h^2 \cdot \frac{f'''(\xi_1) - f'''(\xi_2)}{2}$$

$$= \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

- Two-sided difference (x−h, x, x+h)
- Second order difference scheme for derivative

13

Example: $f(x) = e^x$, $f'(x)\big|_{x=1} = e^x\big|_{x=1} = \boxed{e}$ exact solution

$$\begin{cases} D_1 = \dfrac{f(1+h) - f(1)}{h}, \\ E_1(h) = \underbrace{f'(1)}_{\text{exact soln}} - \underbrace{D_1(h)}_{\text{one-sided diff. approx.}} \end{cases} \qquad \begin{cases} D_2 = \dfrac{f(1+h) - f(1-h)}{2h} \\ E_2(h) = \underbrace{f'(1)}_{\text{exact soln}} - \underbrace{D_2(h)}_{\text{two-sided diff. approx.}} \end{cases}$$

Computational results on Tables 2.1 and 2.2 on p.43 (Single precision)

− Tab 2.2 ⇒ $h \to \frac{h}{2}$, $E(h/2) = \frac{1}{4} E(h)$

TABLE 2.1 Example of Derivative Approximation to $f(x) = e^x$ at $x = 1$

h^{-1}	$D_1(h)$	$E_1(h) = f'(1) - D_1(h)$	$D_2(h)$	$E_2(h) = f'(1) - D_2(h)$
2	3.526814461	−0.8085327148E+00	2.832967758	−0.1146860123E+00
4	3.088244438	−0.3699626923E+00	2.746685505	−0.2840375900E−01
8	2.895481110	−0.1771993637E+00	2.725366592	−0.7084846497E−02
16	2.805027008	−0.8674526215E−01	2.720052719	−0.1770973206E−02
32	2.761199951	−0.4291820526E−01	2.718723297	−0.4415512085E−03
64	2.739639282	−0.2135753632E−01	2.718391418	−0.1096725464E−03
128	2.728942871	−0.1066112518E−01	2.718307495	−0.2574920654E−04

TABLE 2.2 Illustration of Rounding Error in Derivative Approximations, Using $f(x) = e^x$, $x = 1$

h^{-1}	$D_2(h)$	$E_2(h) = f'(1) - D_2(h)$	$E_2(h/2)/E_2(h)$
2	2.832967758	−0.1146860123E+00	NA
4	2.746685505	−0.2840375900E−01	4.038
8	2.725366592	−0.7084846497E−02	4.009
16	2.720052719	−0.1770973206E−02	4.001
32	2.718723297	−0.4415512085E−03	4.011
64	2.718391418	−0.1096725464E−03	4.026
128	2.718307495	−0.2574920654E−04	4.259
256	2.718292236	−0.1049041748E−04	2.455
512	2.718261719	0.2002716064E−04	−0.524

14

- What's wrong for $h = \frac{1}{256}$ that $\frac{E_2(h/2)}{E_2(h)} = 2.455 \not\approx 4$?

To estimate the error, we considered only "mathematical error" only.
Now we consider also the "rounding error". (~ means computed results)

$$f'(x) - \tilde{D}_2(h) = f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x-h)}{2h}$$

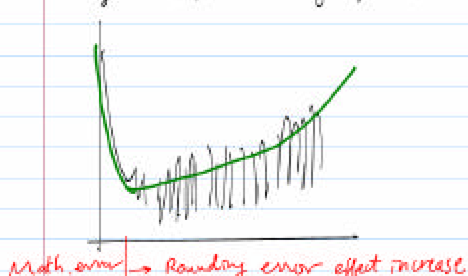
$$= f'(x) - \frac{f(x+h) - f(x-h)}{2h} + \frac{f(x+h) - f(x-h)}{2h} - \frac{\tilde{f}(x+h) - \tilde{f}(x-h)}{2h}$$

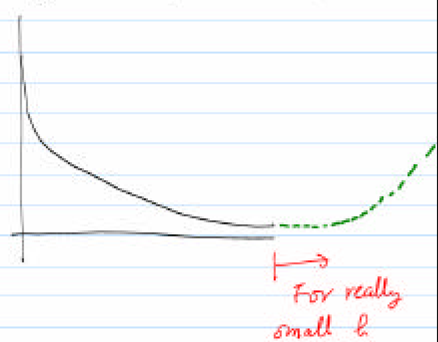
$$= \underbrace{-\frac{1}{6}h^2 f'''(\xi)}_{\text{Math. error due to approximation (A)}} + \underbrace{\frac{f(x+h) - \tilde{f}(x+h)}{2h} + \frac{f(x-h) - \tilde{f}(x-h)}{2h}}_{\text{Error due to rounding (B)}}$$

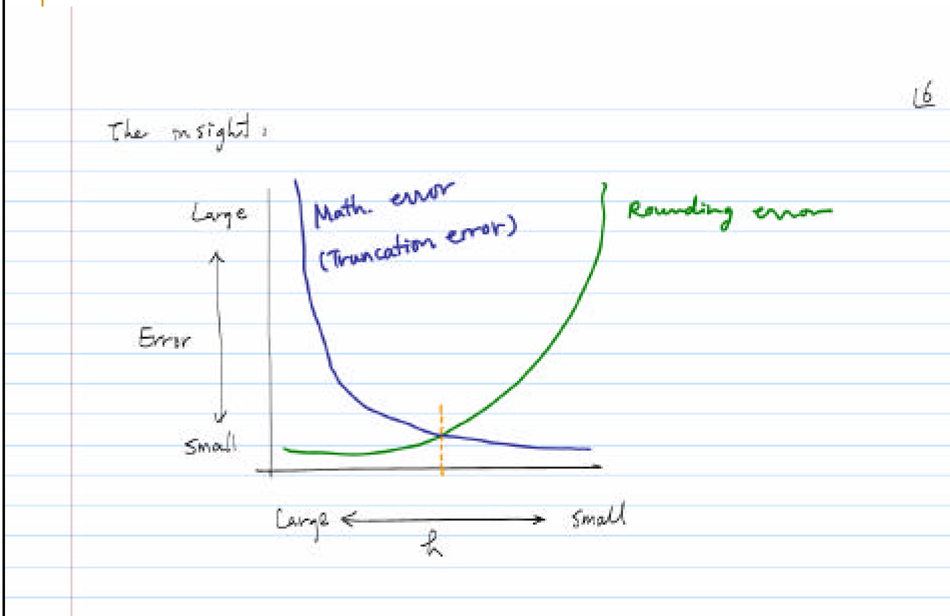
15

For (B), we have $\frac{f(x+h) - \tilde{f}(x+h)}{2h}$ → small rounding err / becoming small ⇒ $\frac{s}{s}$ may be large.

Fig 2.1 (P.44) Single prec. Fig 2.2 (P.46) Double prec.







Find a balance point.

Euler's Method for Initial Value Problem (IVP)

§ 2.3 Euler's method for initial value problems

- Initial value problems for ordinary differential equations.

$$\begin{cases} y' = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

- Use the finite difference to approximate y'.

$$\frac{y(t+h) - y(t)}{h} = f(t, y(t)) + \underbrace{\frac{1}{2} h y''(t_h)}_{O(h)}$$

$$\Rightarrow y(t+h) = y(t) + h \cdot f(t, y(t)) + O(h^2)$$

- Numerical algorithm: (1) $t_n = t_0 + nh$ (t: grid, h: grid size)
  (Euler's Method) (2) $y_{n+1} = y_n + h f(t_n, y_n)$
  (3) Go to step (1)

L2

Example 2.2 (p. 50)

$$\begin{cases} y' = -y + \sin t \\ y(0) = 1 \end{cases} \Rightarrow \text{exact soln: } y(t) = \frac{3}{2}e^{-t} + \frac{1}{2}(\sin t - \cos t)$$

$\Rightarrow$ Euler's iteration: $y_{n+1} = y_n + h(-y_n + \sin t_n)$

- See Fig 2.4 on p.51

DIY – Write a Matlab codes generating similar results in Table 2.5, 2.6 & Fig 2.4.

WAKE UP!!

Newton's Method

§ 3.2 Newton's Method

17

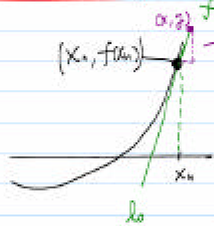
- A classic algorithm for root finding
- An application of the derivative of a function (f')
- Used by Newton in 1669 (330 years ago)
- Even earlier by Joseph Raphson. (Year?)
- Really early by Babylonians. (Year?)

18

Main idea I: Geometric

– Use the tangent line approximation to the function f at the point $(x_n, f(x_n))$

① (Replace a general function by a simpler function.) $l_0(x)$



$$\frac{y - f(x_n)}{x - x_n} = f'(x_n)$$

$$y = f(x_n) + f'(x_n)(x - x_n) \to l(x)$$

⇒ Root of $l_0(x)$ ≡ $y = 0$

⇒ $f(x_n) + f'(x_n)(x - x_n) = 0$

② Solve the simpler function

⇒ $x = x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$

19

Main idea II: Analytical

By Taylor's thm. (who else?)

$$f(x) = f(x_n) + (x - x_n) f'(x_n) + \frac{1}{2}(x - x_n)^2 f''(\xi_n), \quad \xi_n \text{ is btw } x, x_n$$

⇒ $f(x) = 0$

$$\Rightarrow \quad x = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{1}{2}(x - x_n)^2 \frac{f''(\xi_n)}{f'(x_n)}$$

Newton's iteration — Reminder

(Newton's iteration: simpler; whole: original)

Given an expression in terms of something simple plus a reminder, generate a numerical approximation by dropping the reminder term.

One thing can be viewed from several aspects in many cases.

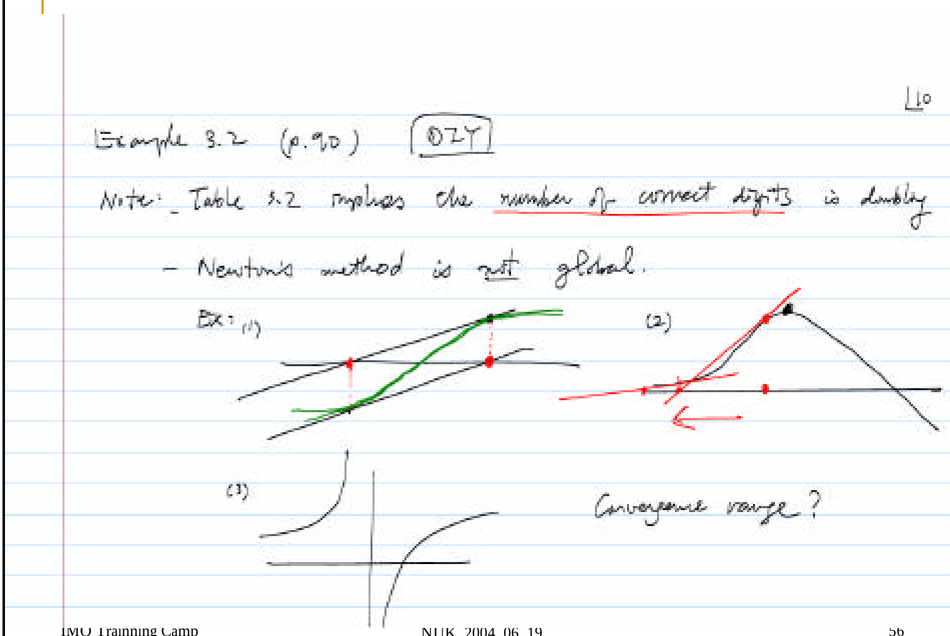


TABLE 3.2 Newton's Method for $f(x) = 2 - e^x$.

n	x_n	$\alpha - x_n$	$\log_{10}(\alpha - x_n)$
0	0.000000000000	0.693147180560	−0.1592
1	1.000000000000	0.306852819440	−0.5131
2	0.735758882343	0.042611701783	−1.3705
3	0.694042299919	0.000895119359	−3.0481
4	0.693147581060	0.000000400500	−6.3974
5	0.693147180560	0.000000000000	−13.0961

[1]

Two lines summary of Newton's method:

- If f, f', and f'' are all continuous near the root, and f' does not equal zero at the root, then Newton's method will converge whenever the initial guess is sufficiently close to the root.
- Moreover, the convergence will be very rapid, with the number of correct digits roughly doubling each iteration.

§ 3.3 How to stop Newton's Method

112

Ideally (but not possible): small error $|\alpha - x_n|$ (α: unknown)

Goal: Rewrite $|\alpha - x_n|$ as a computable quantity.

(I) The Mean Value Thm (L.P.)

$\Rightarrow f(\alpha) - f(x_n) = f'(c_n)(\alpha - x_n)$, c_n is between α & x_n

(II) $\alpha - x_n = \dfrac{f(\alpha) - f(x_n)}{f'(c_n)} = -\dfrac{f(x_n)}{f'(c_n)}$ ($0 = f(\alpha)$)

$= -\dfrac{f(x_{n-1})}{f'(x_{n-1})} \cdot \dfrac{f(x_n)}{f(x_{n-1})} \cdot \dfrac{f'(x_{n-1})}{f'(c_n)}$

$= (x_n - x_{n-1}) \cdot C_n$

Newton's Method: $x_n = x_{n-1} - \dfrac{f(x_{n-1})}{f'(x_{n-1})}$

If $x_n \to \alpha$ $\Rightarrow \lim_{n\to\infty} C_n = 1$ (HW 3.1)

(III) $|\alpha - x_n| = |x_n - x_{n-1}| C_n \to |x_n - x_{n-1}|$ as $n \to \infty$ if converges

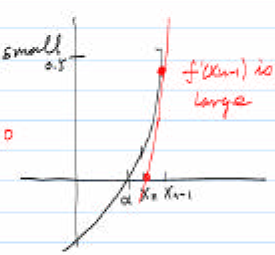
113

Therefore, a possible stopping criterion is

- $|x_n - x_{n-1}|$ is small
- or $|x_n - x_{n-1}| \le \varepsilon$
- or $c\,|x_n - x_{n-1}| \le \varepsilon$

However, it is possible that $|x_n - x_{n-1}|$ is small but x_n is not close to α!

Ex: $|x_n - x_{n-1}| = \left| -\dfrac{f(x_{n-1})}{f'(x_{n-1})} \right|$ — $f(x_{n-1})$: small but not close 0; $f'(x_{n-1})$: BIG; quotient: small



Improvement:

or $\underbrace{|f(x_n)|}_{A} + \underbrace{|x_n - x_{n-1}|}_{B} \le \varepsilon/5$ Both A and B must small enough.

Even you don’t know the answer, you may know what the error is (approximately).

Think globally, act locally

Boundary Value Problem (BVP)

§ 6.10 Boundary Value Problem (21

Let it fly v.s. Pin it down
(IVP) (BVP)

Problem class (I)
Linear BVP

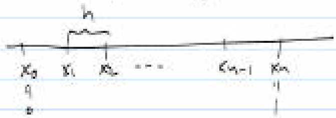
$$\begin{cases} -u'' + p(x)u' + q(x)u = f(x) \\ u(0) = g_0 \\ u(1) = g_1 \\ 0 \le x \le 1 \end{cases}$$

Problem class (II)
Nonlinear BVP

$$\begin{cases} -u'' = F(x, u, u') \\ u(0) = g_0 \\ u(1) = g_1 \\ 0 \le x \le 1 \end{cases}$$

Assumption: (1) Unique solution exists
(2) The solution is smooth

§6.10.1 Simple difference methods



$x_k = k \cdot h$

Taylor's polynomial suggests that

$$\begin{cases} u' = \dfrac{u(x+h) - u(x-h)}{2h} + \dfrac{1}{6} h^2 u'''(\xi_{x,h}) \\ u'' = \dfrac{u(x-h) - 2u(x) + u(x+h)}{h^2} + \dfrac{1}{12} h^2 u^{(4)}(\eta_{x,h}) \end{cases}$$

substitute

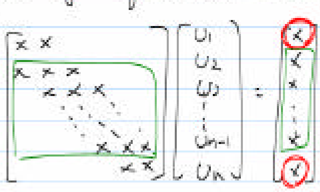
The BVP

$$-u'' + p(x) u' + q(x) u = f(x)$$

$$\Rightarrow -\left(1 + \tfrac{1}{2} p(x) h\right) u(x-h) + \left(2 + q(x) h^2\right) u(x) - \left(1 - \tfrac{1}{2} p(x) h\right) u(x+h)$$

$$= h^2 f(x) + \left(\tfrac{1}{12} h^4 u^{(4)}(\eta_{x,h}) + \tfrac{1}{6} p(x) h^4 u'''(\xi_{x,h})\right) \equiv R(x,h) = O(h^4)$$

The above discretization leads to a tridiagonal linear system defined by Eqs (6.94) - (6.96)



Matlab exercise: (Your code should allow general p(x), q(x).)

Solve the BVP in Example 6.23 by Matlab.

(1) User should be able to input the grid point number n.

(2) Construct the tridiagonal matrix

(3) Solve the linear system.

(4) Plot the solution curve.

the accuracy of the method

125

Thm 6.9 Under the same hypotheses as in Thm 6.8, $\exists C > 0$, independent of h, s.t.

$$\max_{1 \le i \le n-1} |u(x_i) - U_i| \le C h^2 (\|u^{(4)}\|_\infty + \|u'''\|_\infty)$$

Matlab

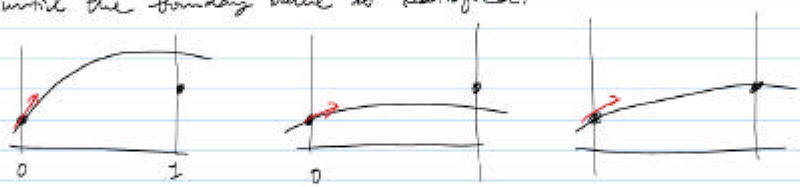
Use your code to reproduce the results in Table 6-14 of Example 6.24 on P.P. 386-387.

126

§6-10.2 Shooting Methods

Main Idea:

Change the boundary value problem to initial value problem.
Guess the initial derivative.
Solve the IVP.
Adjust the initial derivative and solve the problem until the boundary value is satisfied.



[27]

BVP:
$$\begin{cases} -u'' = F(x,u,u') \\ u(0) = g_0 \\ u(1) = g_1 \end{cases} \Rightarrow \text{IVP} \begin{cases} -y'' = F(x,y,y') \\ y(0) = g_0 \\ y'(0) = p \end{cases}$$
Need adjustment

The solution of the IVP $y(x,p)$ is depend on p.
We hope that
$$f(p) = y(1,p) - g_1 = 0$$
That is, we want to find the root of $f(p)$.

Regular secant method:
$$p_{k+1} = p_k - \left(\frac{p_k - p_{k-1}}{f(p_k) - f(p_{k-1})}\right) f(p_k)$$

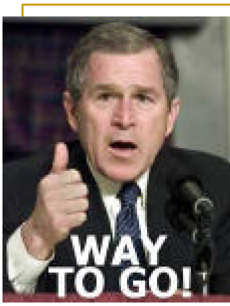
$f(p_i)$ can be obtained by an ODE IVP solver.

[28]

Matlab:
Solve the problem in Example 6.26, P. 389,
$$\begin{cases} -u'' = (u')^2 \\ u(0) = 0 \\ u(1) = 1 \end{cases}$$ with exact solution $u = \ln[(e-1)x+1]$

Let $\begin{cases} p_0 = 1, \\ p_1 = 0.75 \end{cases}$

- In the beginning, we use $h = 1/4$ (coarse grid) to get a better estimation of p.
- Then we use a finer grid $h = 1/128$ to get more accurate solution to p.


WAY
TO GO!


Have Fun!


GOOD LUCK
©RURU


Good Luck!


GOOD LUCK!


Good Luck